UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Attention Guided Image-to-Image translation of Food Images Using GANs

Authors:
Mariya MLADENOVA,
Petar TONCHEV

Supervisor:
Dr. Petia RADEVA,
Javier RODENAS

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

July 1, 2020

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Attention Guided Image-to-Image translation of Food Images Using GANs

by Mariya MLADENOVA,
Petar TONCHEV

Nowadays, with the help of the novel machine learning models, in particular Generative Adversarial Networks, we are able to generate synthetic media, which look absolutely realistic and at the same time authentic. Still, the food image-to-image translation remains a challenging problem that is very unexplored. Due to the complexity of food images the state of the art results are noisy and slow in convergence. In our work, we explore how adding attention to the image-to-image translation on food data can produce more realistic synthetic images and speed up the convergence of the algorithm. Furthermore, we present extensive analysis of GANs for food image synthesis and discuss several possible improvements over the base methodology sharing our insights on this problem.

The source code that has been used to produce the results in this project can be found in our GitLab repository: `https://gitlab.com/deep-food-ub/food-gan`.

Acknowledgements

First, we would like to express our gratitude to our supervisors Petia Radeva and Javier Rodenas for the continuous support of our Master Thesis Project, for their encouragement, patience and expertise in the subject which pushed us to do this exciting project.

Our sincere thanks also goes to Bhalaji Nagarajan and Pau Li Lin for their time and effort helping us with enthusiasm for the project, insightful comments and answers to hard questions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Motivation

In the past few years, there is a huge interest in the image-to-image translation and style transfer. However, most of the work done in this area is concentrating on images from visually similar domains, while very few experiments involve generating synthetic food images. This is quite complex problem due to the inherent complexity of the data. A research in this field, of solving more complex translation tasks, will contribute in the overall improvement of synthetic image generation. Such an enhancement has the potential to enable more applications for translating between diverse domains.

Moreover, a lot of interesting concepts may benefit from the successful photo-realistic translation between food categories. An intriguing application is to unlock new virtual reality (VR) food experiences and their impact on gustatory sensations. Recent topic of interest is vision-induced gustatory manipulation by visually changing one type of food into another and its impact on the gustatory sensations. According to [Nakano et al., 2019a], studies show that it is possible to change the type of food that people believe they are eating by changing its appearance. This happens with a real time image-to-image translation between the actual food the person is eating and the desired dish, that is visualized by the VR set. This idea would help a lot of people, who avoid high-calorie food and have a specific dietary restriction. This includes people, who want to lower their weight, have some food allergies, or a disease like diabetes.

Furthermore, there is a growing interest in computational food analysis due to its importance on health and well-being. Applications include extracting information about the type of food, its ingredients and calories [Lu, 2016], [Bolaños, Ferrà, and Radeva, 2017], [Aguilar et al., 2018]. The techniques tackling these problems often rely on deep convolutional neural networks which need a lot of data to generalize well. Therefore, a successful image translation could be used to artificially expand the current food datasets with synthetic photo-realistic images. This has a great potential as a data enrichment technique, as it generates completely authentic and unseen images, as compared to normal data augmentation of just flipping and rotating the images. This method adds a larger diversity of the classes, moreover it includes diverse backgrounds of different environments in the data, which does help solving the problem of recognizing food in images with complex background.

1.2 Problem Statement

Food images are challenging to work with as they have a very large variability within the classes - you can present the same dish in many different ways and shapes

FIGURE 1.1: Example of variability in a food class: All images are showing a chicken curry. The way it is cooked and presented completely changes the look of the meal.

(Figure: 1.1). In the same time two completely different meals can look the same, again depending on the presentation (Figure: 1.2). Lately, a lot of advancements have been made on image generation tasks using Generative Adversarial Networks. Most of domain translation GAN architectures are tested of human faces or horses and zebras. In such datasets the source and target domains have a lot of visual overlap similarity and the actual problem is to change the colour, shape or texture, but not all of them at once. However, generating synthetic food images and translating from one category to another remains a very challenging problem due to its diverse form, colour, shape, texture and dish presentation. Therefore, food category translation model needs to be able to deal with all of the different types of changes at once.

In this project we tackle the problem of generating synthetic photo-realistic food images starting from one food class and targeting another. We are using attention guided Generative Adversarial Networks (GANs), having a cycle-consistency constraint architecture, in particular AttentionGAN [Tang et al., 2019]. This model has several advantages, which are in help for the food image-to-image translation. First of all, the cycle-consistency constraint allows us to use unpaired dataset to train the model, at the same time the attention mechanism improves the final generation, by guiding the generator to apply changes only to the desired part of the image, leaving the background untouched. However, sometimes the generated attention is not satisfying, therefore we propose a possible improvement of the original model, by integrating a salient object detection algorithm PAGE-Net [Wang et al., 2019b].

In the next section we discuss the state of the art methods applied in this field. After this, we explain in details the architectures of the AttentionGAN and the PAGE-Net. In the Experiments section we explain the experiments we did, followed by results, evaluations and comparison between different models. We make a conclusion and discuss possible future improvements in the last chapter.

FIGURE 1.2: Example of similarity between classes in a food class: Here, two completely different meals - baklava and lasagna are looking quite similar.

Chapter 2

SoTA on GANs and Food Image Analysis

Generative Adversarial Networks are proven to be powerful technique for generating photo-realistic images [Brock, Donahue, and Simonyan, 2018], style transfer [Li et al., 2017] and image-to-image translation [Isola et al., 2017]. In their work [Tanno et al., 2018] apply for the first time GANs to food data in order to translate between 10 kinds of food that are served in a bowl-type dishes. They perform image-to-image translation based on the CycleGAN [Zhu et al., 2017] which allows them to train the model with unpaired images. However, to enable the multi-domain translation, they extend the base CycleGAN architecture by adding an auxiliary loss term in the discriminator similar to the StarGAN [Choi et al., 2018] approach. Their approach shows that it is possible to generate decent photo-realistic food images (Figure: 2.1) when trained on a huge dataset of 230,000 images. Moreover, they show that the size of the dataset makes a lot of impact on the quality of the generated images. Similar approach on the same dataset is applied in [Nakano et al., 2019a] and [Nakano et al., 2019b] which directly use StarGAN [Choi et al., 2018] for real-time food category translation on VR set in order to investigate the impact of a GAN-based system on gustatory sensations and food recognition. More recently, the so-called PizzaGAN [Papadopoulos et al., 2019] has presented very unique way of generating realistic food images by employing composable module operations which are able to add or remove particular ingredients from an image of pizza. These operations are doing image-to-image translation based on CycleGAN and mimic the step-by-step procedure of cooking a pizza. The algorithm was trained on a reasonable large dataset of more than 9000 images.

Another advancement in the field of food computation is the ability to learn cross-modal embeddings between recipes and corresponding food images [Salvador et al., 2017]. This led to an alternative approach of generating synthetic food images by translating from textual descriptions of recipes into their corresponding visual representation. In their work [Sabini, Abdullah, and Phan, 2018] the authors input the learned recipe embeddings into a DCGAN. Later, Adversarial Cross-Modal Embeddings (ACME) [Wang et al., 2019a] beats state of the art results on food recipe retrieval by incorporating WGAN-GP in order to align the distribution of the embeddings from the different domain. As a by-product of their architecture, they show it can generate food images. More recently, CookGAN [Han, Guerrero, and Pavlovic, 2020] tackles the same problem of generating photo-realistic food images from their recipes. They show that using StackGAN++ [Zhang et al., 2018] with added cycle consistency constraint, they are able to generate photo-realistic food images (Figure: 2.3). As already mentioned, all these techniques are trained on the huge dataset Recipe1M [Marin et al., 2019] which contains 1 million different recipes with at least one image per recipe.

FIGURE 2.1: Translating between 10 bowl-type food categories images using the extended CycleGAN architecture in [Tanno et al., 2018]. The leftmost images are input and the others are generated with respect to the corresponding category (Preprint [Tanno et al., 2018]).

As far as as we know, these are all the works that perform some kind of synthetic food generation. A major limitation in all of the above works is that they require using huge datasets. The recent advancements in deep learning has shown that using attention mechanisms can direct the neural networks to focus on regions of interest that are the most relevant for the task. More specifically to the subject of synthetic image generation AttentionGAN [Tang et al., 2019] shows that adding attention to the generator improves the quality of the generated images. However, neither of the current approaches of generating synthetic food images use attention guided mechanisms in their GANs. Therefore, to the extent of our knowledge, we are the first to apply attention guided image-to-image translation in the food domain. Furthermore, another even more challenging contribution is the use of a small-scale dataset of just 1500 training images on this problem.

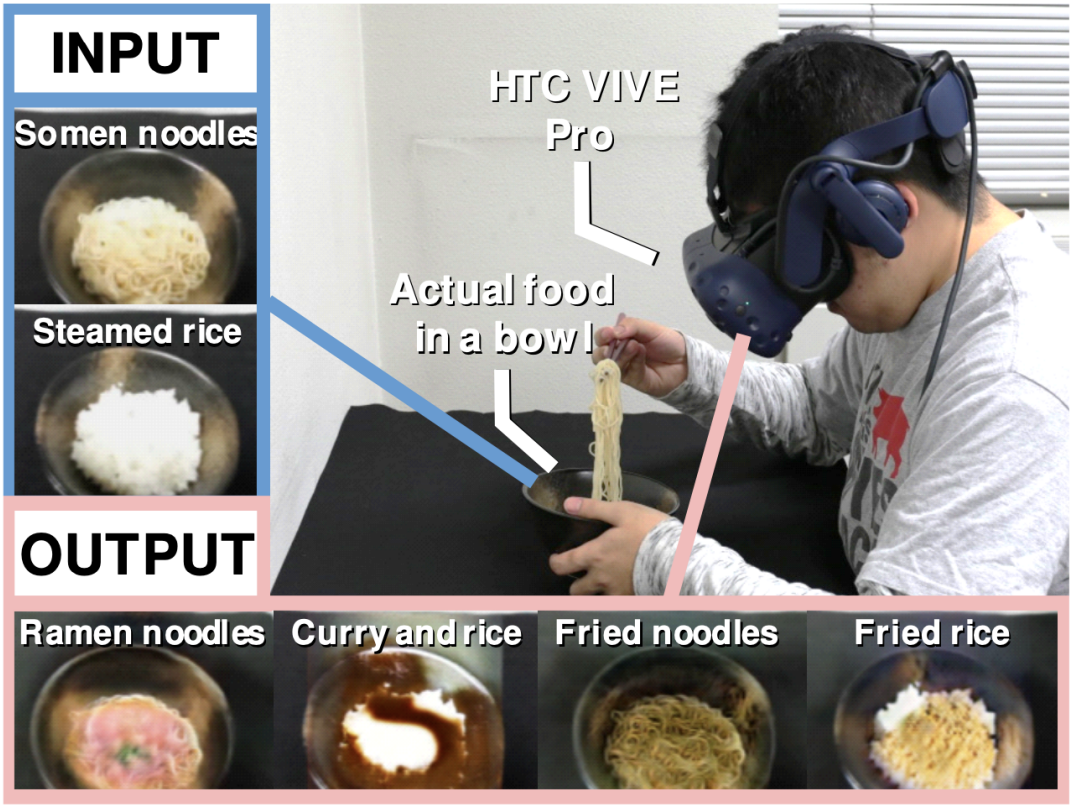


FIGURE 2.2: Real-time food-to-food translation system. Somean noodles or steamed rice is taken as an input and they get translated to ramen noodles, curry and rice, fried noodles or fried rice. (Preprint [Nakano et al., 2019b]).

FIGURE 2.3: Example results on three different ingredients using the recipe-to-image strategy. Ours is referred to CookGAN. (Preprint [Han, Guerrero, and Pavlovic, 2020]).

Chapter 3

Methodology

3.1 Background on GANs and Attention

In the recent years, Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] have been a very hot topic for the researchers in the Deep Learning field. The idea behind GANs is to train two models Generative Model and Discriminative Model in a framework that corresponds to a two-player mini-max game. The goal of the generator (G) is to create a new plausible samples that are able to fool the discriminators (D) that they are part of a real dataset. At the same time the discriminator aim is to correctly identify the fake samples. This is defined by the following loss function:

$$L_{GAN} = \min_{G} \max_{D} [log(D(x)) + log(1 - D(G(z)))], \tag{3.1}$$

where x is a real image from the training set and z just noise, sampled from a prior noise distribution. The result is a powerful generative model that produces impressive results on image generation tasks.

Conditional GANs [Mirza and Osindero, 2014] build upon the basic GANs by putting some restrictions y on the generated image in order to make them meet the user requirements. In this case the generator is receiving as an input not only noise, but together with the additional information y. The loss function of conditional GANs is defined by:

$$L_{CGAN} = \min_{G} \max_{D} [log(D(x|y)) + log(1 - D(G(z|y)))]. \tag{3.2}$$

The most basic types of image-to-image translation with GANs take two paired images that have one-to-one correspondence and learn to translate from one to the other [Isola et al., 2017]. This method is called paired translation. However, the main limitation of similar architectures is that one can rarely find paired set of images from the source to the target domain. Thus, image-to-image translation is especially on hard problems like food image-to-image translation, where no such dataset exist and perhaps is impossible to build.

To tackle this issue, a cycle consistency constraint was adopted to learn the translation between domains without paired images [Zhu et al., 2017]. The idea is simple - if we translate a sentence from English to French and then back to English we should obtain the exact same sentence we started with. In the GAN setting we have a translator G that translates from X to Y and another translator F that translates Y

to X, so they are in a sense inverse of each other. Therefore,

$$F(G(x)) \approx x \text{ and } G(F(y)) \approx y. \tag{3.3}$$

So when both mappers are trained simultaneously by combining this cycle consistency loss with an adversarial losses on domains X and Y would produce a framework for unpaired image-to-image translation. While there exist other architectures that tackle the same problem, most of them can be distracted by the background of the input images, thus, failing to focus on the important parts of the image.

Recently, attention mechanisms have been employed in a lot of machine learning models. The idea behind attention mechanisms initially comes from natural language processing field and more specifically, neural machine translation using sequence to sequence models. The problem there was that the short memory of the system was not able to process long sequences. Attention mechanisms deal with this problem by using the intermediate parts of the sequence, to help the translation process. By using these intermediate vectors, the algorithm can learn which of them are more important in a specific state of the translation, hence develop more attention on them.

Similarly, attention mechanisms have been integrated into convolutional neural networks for finding attention regions in images, which help the classification process [Zagoruyko and Komodakis, 2016].

Attention-Guided Translation is a novel method, which uses attention mechanism introduced into image-to-image translation, which not only avoids the need of a paired data sets, but moreover, it finds the foreground of the images and apply generated changes only on these parts, while leaving the background untouched.

3.2 Attention GAN

AttentionGAN [Tang et al., 2019] is a novel method, which applies attention-guided GANs for unpaired image-to-image translation. The results shown on the original paper are really promising. However, they have tested it on completely different fields from food, like horse to zebra, facial expression or selfie to anime translations. Still, their results shown on the paper are the most realistic ones, compared with CycleGAN [Zhu et al., 2017], StarGAN [Choi et al., 2018], IcGAN [Perarnau et al., 2016e], GANimorph [Gokaslan et al., 2018] and other SoTA methods.

The AttentionGAN has two generation schemes proposed:

- Scheme I (Figure: 3.1)

  The scheme I generator has a build in attention module, which produces an attention mask, which identifies the foreground from the background, and a content mask, which is simply the generator image output, if there was no attention module. After that, the input, the attention mask and the content masks are fused together to create a generated image, where changes are applied only on the foreground of the image and the background is taken from the original one. In scheme I, they have developed as well, an attention-guided discriminator, which considers only the changed foreground regions, not the whole image.

  However, in the original paper, the authors conclude, that this method works only on translations with minor changes between the source and the target classed, like facial expression translation. In their experiments with horse to zebra, apple to orange or map to satellite photo translations, this scheme does

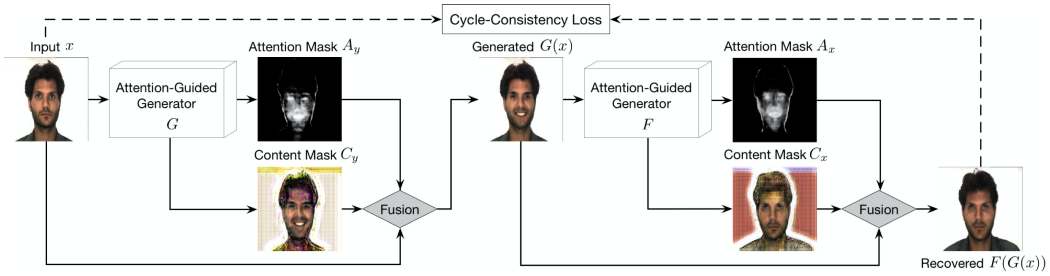


FIGURE 3.1: The architecture consists of two attention guided generators, one of which translates an image from class A to image of class B and the other one, which translated the fake image from class B again to class A image. The original and the final images are then compared by the cycle-consistency loss. The generators have a build-in attention modules, each of which generates one attention mask an one content mask per image. (Preprint [Tang et al., 2019]).

FIGURE 3.2: Comparison between two schemes in different types of translations. (Preprint [Tang et al., 2019]).

not apply any significant change in the source image (Figure: 4.6) Therefore, we will not describe in details the formalization of this architecture and will focus our attention to scheme II.

- Scheme II (Figure: 3.3)

  The scheme II generation architecture is more complex than the scheme I. Here, the generator has two separate sub-nets, one of which is responsible for generating attention masks, and the other one - content masks. They are completely separate and have their own network parameters. Other major difference is that here, there is not only one attention mask, and only one content mask, but the attention generator produces $n-1$ foreground attention masks and one background mask. For each of these foreground attention masks, a separate content masks is generated by the content generator. In this way the generation space is enlarged from 3-channel space, to $3n$-channel space. This allows the model to learn way more complex image-to-image translations. More detailed formalization is described in the next subsection.

  This method has a lot better results in different translations as seen in Figure 4.6. As we need to translate food images, which have very large variety in shape, texture and colour, we decided to use scheme II in our experiments.

3.2.1 AttentionGAN scheme II

Attention-Guided Generation

As described above, a generator G for an image to image translation from image x to image y, consists of two sub-nets having a parameter-sharing encoder G_E, an attention mask generator G_A and a content mask generator G_C. G_A produces $n-1$ attention masks $\{A_y^f\}_{f=1}^{n-1}$ and one background mask A_y^b using a channel-wise softmax function as a normalization, while G_C generates $n-1$ content masks $\{C_y^f\}_{f=1}^{n-1}$. Finally they are combined in the following way:

$$G(x) = \sum_{f=1}^{n-1}(C_y^f * A_y^f) + x * A_y^b \tag{3.4}$$

Similarly the generator F does the opposite:

$$F(y) = \sum_{f=1}^{n-1}(C_x^f * A_x^f) + y * A_x^b \tag{3.5}$$

Attention-Guided Cycle

The idea behind the unpaired image-to-image translation is the cycle-consistency, which is translated to the following expression: $x \to G(x) \to F(G(x)) \approx x$, meaning that if we translate an image from one class to another and then translate again to the source class, the result should be approximately the same as the source image. Similarly, the other way around $y \to F(y) \to G(F(y)) \approx y$. This implies the cycle-consistency loss:

$$L_{cycle}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_{data}(y)}[\|G(F(y)) - y\|_1] \tag{3.6}$$

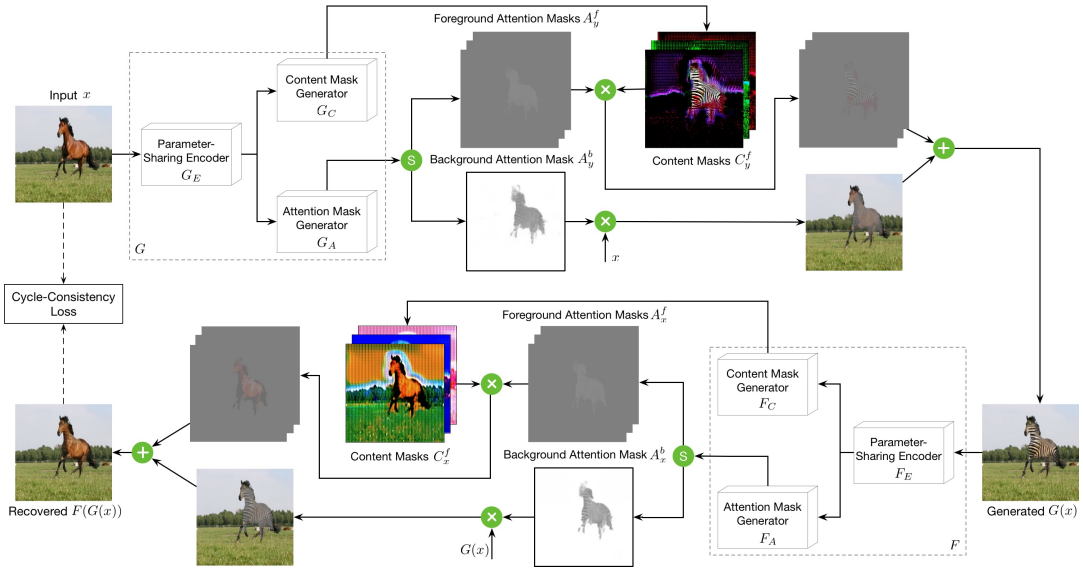


FIGURE 3.3: In this scheme, each of the generators have a parameter-sharing encoder, an attention mask generator and a content mask generator. The attention generator generates $n-1$ foreground attention masks and one background attention mask. The content generator produces $n-1$ content masks. Each foreground attention mask is multiplied by its corresponding content mask, while the background attention is multiplied by the original input image. After that the results from the multiplication of attentions and content masks are summed with the result of the background attention with the input image. This produces the final generated image. Again, in this scheme there is a cycle consistency loss, which compares the result from the original image, with the recovered image, once the image go through both generators and finish the whole cycle. (Preprint [Tang et al., 2019]).

Attention-Guided Discriminator

The discriminator's function is to learn to distinguish between a real and a fake image, more specifically the discriminator D_Y takes as input the fake image $G(x)$ and classifies it as either real or fake, while using the adversarial loss as an optimization function. The adversarial loss looks as follows:

$$L_{GAN}(G, D_Y) = \mathbb{E}_{y \sim p_{data}(y)}[logD_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[log(1 - D_Y(G(x)))] \tag{3.7}$$

Similarly, the optimisation function of the discriminator D_X is:

$$L_{GAN}(G, D_X) = \mathbb{E}_{x \sim p_{data}(x)}[logD_X(x)] + \mathbb{E}_{y \sim p_{data}(y)}[log(1 - D_X(G(y)))] \tag{3.8}$$

The aim of the discriminator is to maximise this function, while the generator is trying to minimise it. This defines the mini-max game, which the two classifiers are playing. While playing the game, they constantly compete with each other, which leads to improvement in both of them.

Defining the Loss Function

The loss function of this method is defined by the summation of the adversarial loss, the cycle-consistency loss and and identity preserving loss, while the last two are scaled by some λ parameters. That is:

$$L = L_{GAN} + \lambda_{cycle} * L_{cycle} + \lambda_{id} * L_{id} \tag{3.9}$$

3.3 PAGE-Net

A possible improvement of the attention mechanism used in AttentionGAN is Pyramid Attentive and salient edGE-aware saliency model (PAGE-Net) [Wang et al., 2019b]. The method has a really good performance even for images without a distinctive separation, between the foreground and the background, as in the case of food images.

The novelty for this method is that it works on multi-scale, creating a pyramid attention structure, using a convolutional neural networks. On top of this pyramid attention, they have designed a salient edge detection module, to improve the segmentation, creating a more realistic and sharp edge between the salient object and the background.

The architecture of the method consists of three main components - a backbone network for feature extraction, the pytamid attention module and the salient edge detection module.

Pyramid Attention Module

The aim of this method is to find important regions in the image and consider mainly features of these regions on multiple scales. To do so, they have build a stacked attention architecture by stacking multiple attention layers build upon multi-scale features. This module is called *pyramid attention model*.

We define $X \in \mathbb{R}^{M \times M \times C}$ as the 3D feature tensor from the convolutional layer of the saliency network, where C is the number of channels and M is the width and height. Then the aim of the pyramid attention module is to obtain multi-scale

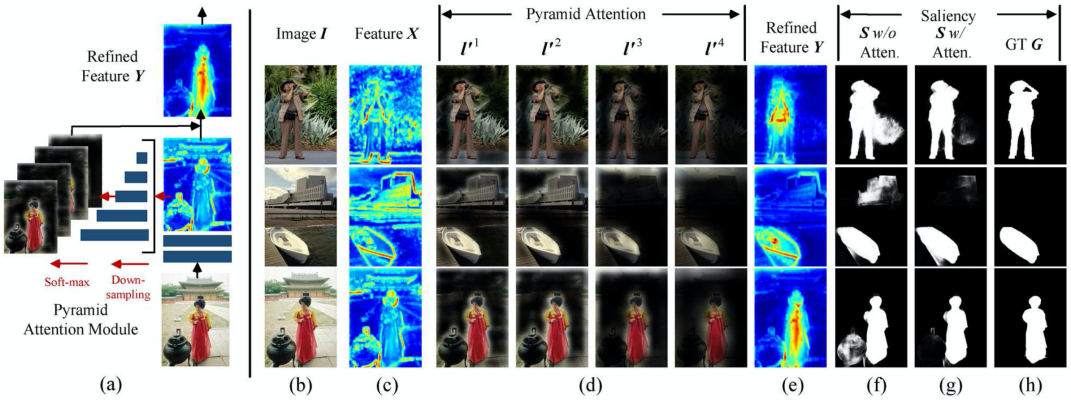


FIGURE 3.4: Illustration of the pyramid attention module: (a) shows the flow of the module, (b) shows the input image, (c) shows the initial features of X, without applying the pyramid attention module, (d) highlights the important regions on each scale, (e) shows the features after applying the attention, (f) shows the result without the attention module, (g) is the result with the attention module and (h) is the ground truth. Clearly the attention module has quite large effect on the saliency recognition (Preprint [Wang et al., 2019b]).

features by reducing the width and height dimensions of X by 2, N times - $X^n \in \mathbb{R}^{\frac{M}{2^n} \times \frac{M}{2^n} \times C}$, where $n = 1, 2, 3, ..., N$. After that for each scale n, a soft attention mechanism is applied to predict an importance map $I \in [0,1]^{\frac{M}{2^n} \times \frac{M}{2^n}}$. In this way the model learns a normalized importance weight (attention map) for each region at every scale. The next step is to up-sample back to the original resolutions - $\{I'^n \in [0,1]^{M \times M}\}_{n=1}^{N}$. These attention maps are used to improve the original feature representation X by accounting for the expectation of the feature slices in different regions. That is:

$$Y_j = \frac{1}{N} \sum_{n=1}^{N} (1 + I_j'^n) X_j, \quad j \in 1, ..., M \times M, \tag{3.10}$$

where Y is the updated feature and Y_j i the j-th slice of the feature cube. The whole process is illustrated in Figure: 3.4.

Salient Edge Detector

The smoothness of the convolutional kernels in the neural network, together with the down-sampling are resulting in really unclear boundary of the detected objects. To deal with this problem, the authors of the paper have been integrated an salient edge detection module to improve the boundaries. The modules consist of a stack of convolutional layers and the following L2 norm loss function:

$$\frac{1}{K} \sum_{k=1}^{K} L^{Edg}(P_k, F(Y_{I_k})), \tag{3.11}$$

$$L^{Edg}(P_k, F(Y_{I_k})) = \|P_k - F(Y_{I_k})\|_2^2, \tag{3.12}$$

where K is the size of the training set, I_k is the color image, P_k is the ground truth salient object boundary map, F is the edge detection module and Y_{I_k} is the enhanced

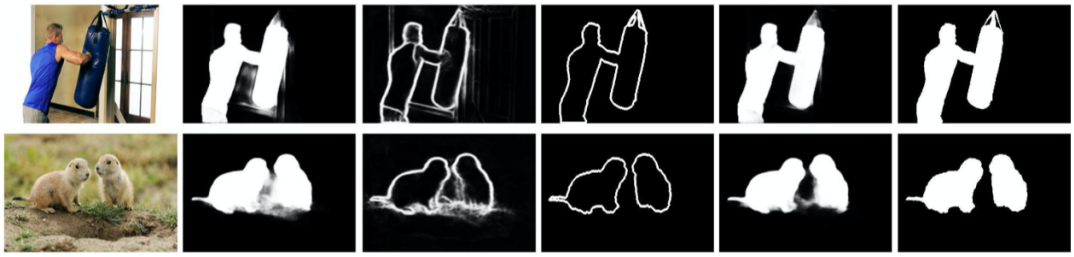


FIGURE 3.5: Illustration of the salient edge detention module: As clearly seen, the edge detection improves a lot the saliency, by sharpening the edges in order to obtain more accurate and realistic boundaries of the objects. (Preprint [Wang et al., 2019b]).

feature of the training image I_k. After having the edge estimation, a saliency readout network $R(Y_{I_k}, F(Y_{I_k}))$ is estimating the saliency, by taking in account both the feature vector Y_{I_k} and the edge estimation $F(Y_{I_k})$, minimizing the following loss function:

$$\frac{1}{K}\sum_{k=1}^{K}(L^{Sal}(G_k, R(Y_{I_k}, F(Y_{I_k}))) + L^{Edg}(P_k, F(Y_{I_k}))), \tag{3.13}$$

$$L^{Sal}(G, R(Y_I, F(Y_I))) = -\sum_i \beta(1-G_i)log(1-S_i) + (1-\beta)G_i log(S_i), \tag{3.14}$$

where G_k is the ground truth saliency map of image I_k and β is the salient pixel ratio of the G. $i \in \Omega_I$, where Ω_I is the lattice domain of image I, S is the saliency estimate of R and $S_i \in S$. Here, L^{Sal} is a weighted cross-entropy loss that accounts for data imbalance between salient and non-salient pixels. An illustration of the salient edge detector result is given in Figure 3.5.

To make use of the information from different layers, they have introduces dense connections in they networks. This means that the feature vector Y^l in the l-th layer is considering all multi-layer saliency estimates and all edge information from the preceding $l-1$ layers:

$$Y^l = [Y^l, H^l(E^{l-1}, ..., E^1, S^{l-1}, ..., S^1)], \tag{3.15}$$

where H is a network that up-samples and concatenates the additional inputs from the preceding layers.

This method has an impressive performance even in images with really complex background, like food images. That is why we decided to experiment integrating it as attention in the GAN architecture. Some examples of how the method performs on food images is shown in Figure 3.6

FIGURE 3.6: Examples of salient recognition of food images.

Chapter 4

Experiments

We conduct extensive experiments to gain insights of food image generation. First, we explore the ability of AttentionGAN to generate synthetic food images. Then, we propose possible improvements over the base AttentionGAN architecture. We describe the setup of our experiments in this chapter.

4.1 Dataset

For image-to-image translation problem we employed the Food101 dataset [Bossard, Guillaumin, and Van Gool, 2014]. It has 101 food classes (Figure: 4.1), and a total of 101k images, where each class consists of 1000 images - 250 testing and 750 training samples. However the dataset is quite noisy. There are a lot of wrong labels and inadequate images (Figure: 4.4).

Following the Food101 data splitting, we use the 750 train samples per category to train the GAN and the others are used for testing. Without loss of generality, we used mostly the spaghetti bolognese (Figure: 4.2) and pizza (Figure: 4.3) classes since such a problem consists of translating the shape, scale, color and texture, while at the same time images from the two classes sometimes share the same round shape and positioning in a dish. Moreover, we also experimented with pizza to cupcake translation in order to investigate performance on a more visually dissimilar categories. The cupcake images (Figure: 4.5) were taken from the "cupcake" category of the Food101 dataset, too.

4.2 Implementation setting

For the training of the models we used a machine with 8 core Intel i7-4770 CPU at 3.40GHz with NVIDIA GeForce RTX 2080ti with 11 Gb memory. The code was written in Python 3 using the PyTorch deep learning library. Due to the GPU memory limitations we were forced to use smaller batch sizes than usual.

On this machine the training of an AttentionGAN takes from 215 to 260 seconds per epoch, depending on the specific architecture we select.

4.3 AttentionGAN Validation

To explore the applicability of the AttentionGAN to food image-to-image translation problem we conducted various experiments on the base architecture and possible improvements by altering the vanilla architecture in a search for more realistic synthetic images, better attention mechanisms and indicators of convergence.

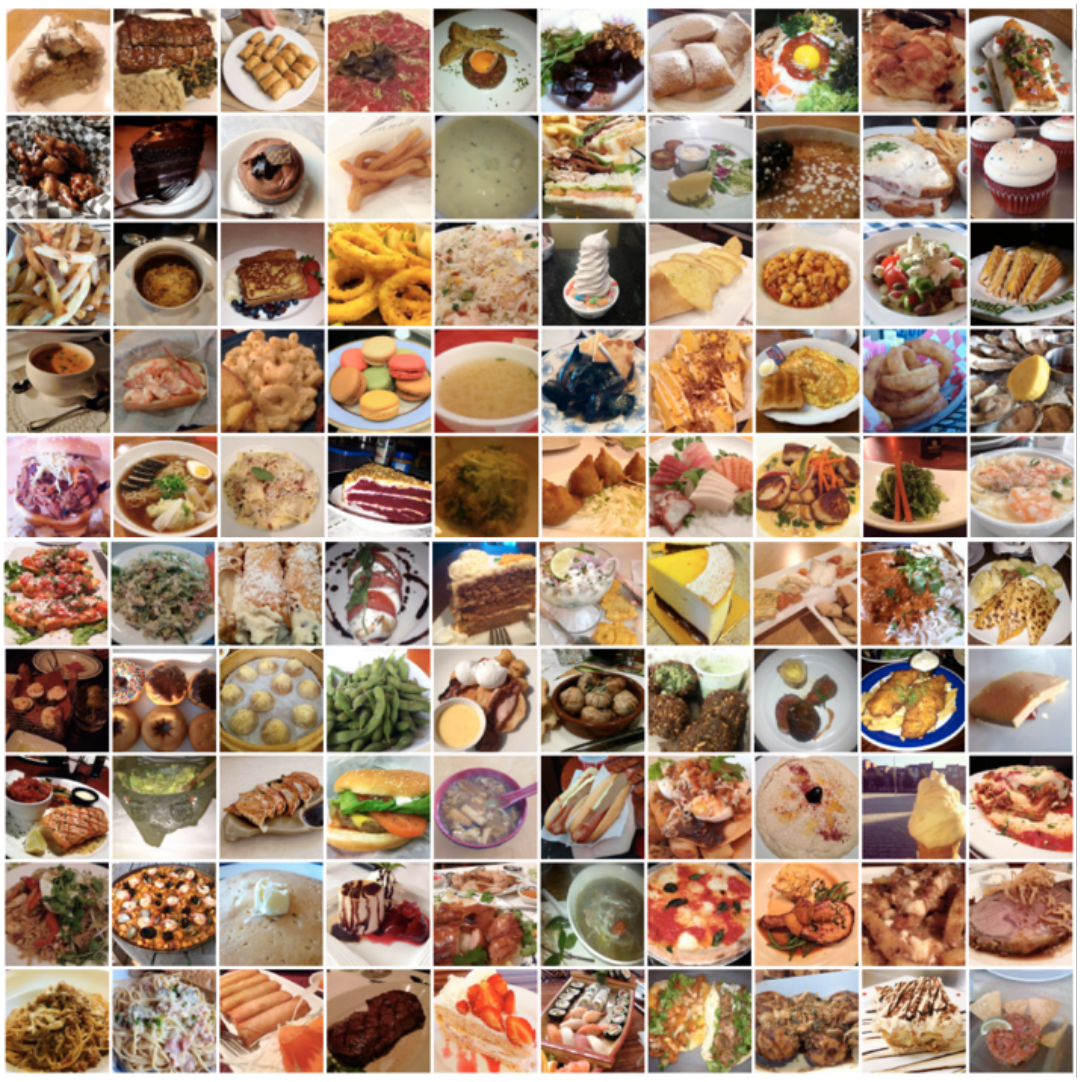


FIGURE 4.1: 100 out of 101 food categories in the Food101 dataset. Image is taken from [Bossard, Guillaumin, and Van Gool, 2014].

FIGURE 4.2: Examples of images in the spaghetti bolognese class

FIGURE 4.3: Examples of images in the pizza class

FIGURE 4.4: Examples of noisy images in the dataset: All of these images are taken from the training set of the pizza class in the Food101 dataset

FIGURE 4.5: Examples of images in the cupcake class

FIGURE 4.6: AttentionGAN scheme I vs scheme II pizza2pasta example: The scheme I is not applying any change to the source image, while the scheme II generates a realistic pasta image.

4.3.1 Parameter setting

We follow the [Tang et al., 2019] parameter settings: we scale the images to 256x256, doing horizontal flips and random crops for data augmentation. The optimizer is Adam with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ momentum terms. The loss scaling parameters are set as following: $\lambda_{cycle} = 10$, $\lambda_i d = 0.5$. Due to the limited GPU memory of 11 Gb we were forced to run AttentionGAN experiments with batch size of 2, rather than the default size of 4.

4.3.2 Vanilla AttentionGAN

First, we started from the unaltered AttentionGAN architecture to obtain a baseline. In the original paper the authors show that scheme I is applicable only on translation tasks that have large overlap similarity in the source and target domain, such as facial expression-to-expression task. In Figure 4.6 we observe, that after trying both schemes to our dataset, the scheme I did not have any effect on the translation, while the scheme II generated a really realistic image. Taking into consideration the complexity of the food data translation problem, we limit our experiments to using scheme II of the AttentionGAN.

By default the architecture needs 60 epochs with a batch size of 4 to start observing a few successful translations from horses to zebras. Due to the complexity of the food translation problem and the smaller batch size that we can use, we found that 60 epochs are not enough and the algorithm must be trained for a lot more. At about 200 epochs it starts translating pizza to pasta successfully on parts of the images that have more visual similarity like cheese to spaghetti and chopped tomatoes to mince (Figure: 4.7). However, to produce more consistent translations across the whole dataset, more epochs are needed.

4.3.3 AttentionGAN with Additional Content Masks

In the original AttentionGAN paper, the authors claim that for some generation tasks when the foreground generation is very difficult (e.g. horses to zebras) it is beneficial to add additional content masks. As stated above, food generation is a

FIGURE 4.7: Translating from pizza to pasta using the AttentionGAN model on the same image generated on different epochs to get a sense how much epochs are needed. At 200 epochs it successfully translates the texture from the source to the target domain in a realistic way.

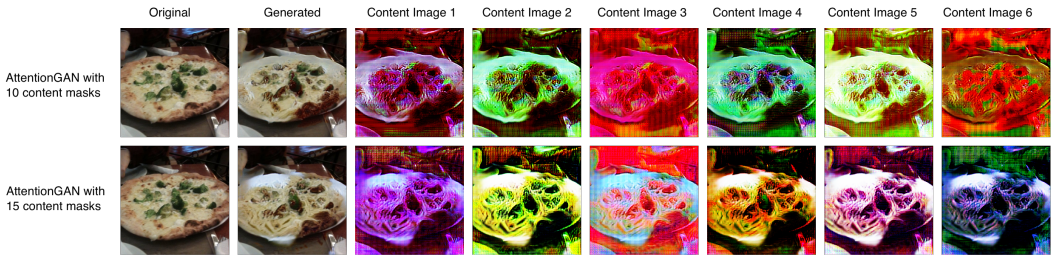


FIGURE 4.8: The figure shows the comparison between the base AttentionGAN with 10 content masks and AttentionGAN with 15 content masks at the 200 epoch. The image shows the first 6 content masks from both models. The additional content masks improve the generated texture and thus make the images more realistic.

very complex problem and as a result of this observation we extended the base architecture by adding 5 more content and foreground attention masks. On Figure 4.8 we observe that the additional content masks help produce better textures. Although, both images are far from perfect since the training is not converged yet we can clearly observe that the additional content masks resulted in a higher detail spaghetti texture. Our initial intuition behind adding more content masks was that when combined they would be able to produce more realistic texture. However, the results show that the additional masks have richer texture on their own compared to the base AttentionGAN. This shows that the extended architecture converges faster than the base one. For simplicity and future reference, we will name this model 'A-GAN 15'.

We found that around 800 epochs are needed for the algorithm to produce consistent and realistic results. At this stage it even starts to change the shape of the food by generating realistic plates.

4.3.4 PAGE-Net Integration

Although, the fully trained AttentionGAN generates good attention masks in our observations most of the failing images of the fully trained models were not due to bad content generation, but due to bad attention masks (Figure: 4.10). Moreover,

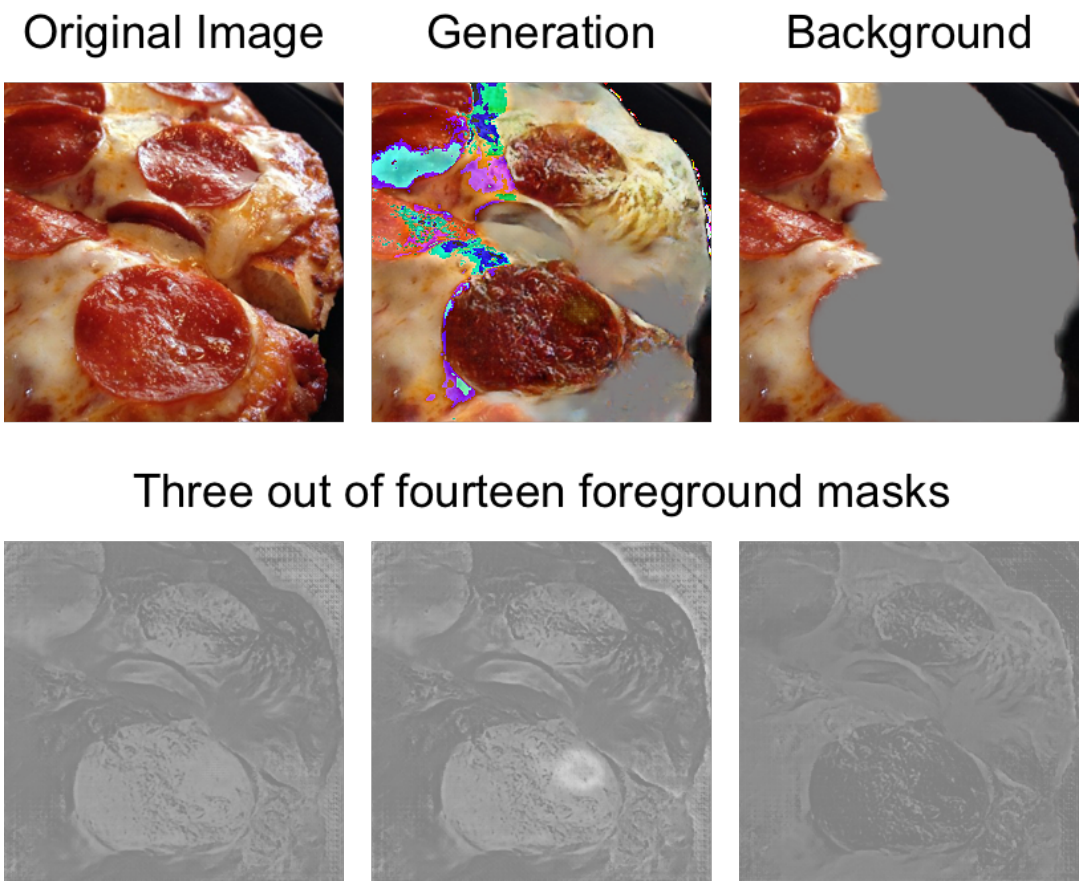


FIGURE 4.9: Example of a bad generation with noise due to overflow of values in the areas which are considered as both a foreground and a background. However, there are green areas in the bottom right corner of the generated images, which is not considered as neither a background, nor a foreground.

the algorithm is not producing any meaningful results for the first 100 epochs since it needs to learn the attention from scratch. To both speed up the training and improve the quality of the attention masks, we decided to integrate a pre-trained attention mechanism that aids the attention generator. The PAGE-Net algorithm [Wang et al., 2019b] demonstrates exceptionally good results on generating attention masks. Its flexibility allows for producing state of the art attention masks on food images even without fine-tuning the algorithm on a food dataset. For this reason, we altered the AttentionGAN to use a background attention mask that is generated from PAGE-Net. We experimented with both applying the PAGE-Net attention only to the background, and to the foreground and the background.

PAGE-Net Attention as a Background Mask

Firstly, we replaced only the background attention mask with the PAGE-Net. However, the experiments was not satisfactory (Figure: 4.9). The fact that different algorithms are producing the background and the foreground masks, results in both overlapping and empty regions in the image. In such situations the values in the overlapping regions might overflow, which produces noise, while the empty regions are left out grey, as there is no content which is included there.

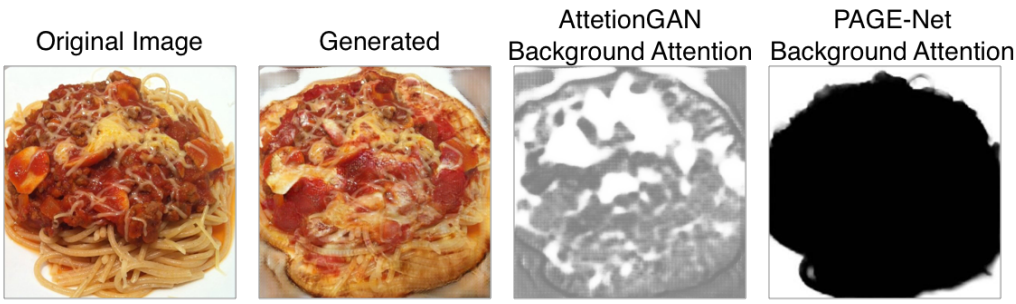


FIGURE 4.10: The generated image here looks quite nice from far away, with dense and textured crust, nice colours and realistically looking toppings and good shape. However if one takes a closer look into the image, a lot of spaghetti shape artifacts appear, which are left from the original image. The reason is the background attention mask which is clearly wrong and quite noisy. According to the AttentionGAN a large part in the middle of the pizza is considered as background, while the white plate is not. A solution for this problem is the rightmost image where it is shown the output of the PAGE-Net algorithm, given the original image. As you can see, the segmentation is really good, a lot better than what the attention mechanism of AttentionGAN has produced.

PAGE-Net Attention as a Background and a Foreground Mask

An experiment we did in order to solve the above problem was to use the PAGE-net as both the background and foreground masks. In this method we created the background mask from PAGE-Net, then we inverted it and used it as a foreground mask for multiplication with each of the fourteen content masks. The foreground masks have a range from 0 to 255, so we needed to scale them with a training parameter γ, so that the values do not overflow after the summation of each foreground mask multiplied by a content mask. Unfortunately, this resulted in again unsuccessful trial. The reason that the noise is still appearing might be due to the fixed γ parameter for all foreground masks.

4.3.5 PAGE-Net Integration with Scaling

To solve the previous problems, we alter the original generator function to multiply each foreground attention by some scalar parameter γ whose value is learned in the training:

$$G(x) = \sum_{f=1}^{n-1}(C_y^f * A_y^f * \gamma_y^f) + x * A_y^b$$

The γ parameter role is to downscale the foreground attention masks on regions where the AttentionGAN foreground attentions and PAGE-Net background attention overlap. In order to be more precise, each foreground attention has its separate γ parameter. In order to avoid overflows we must restrict the sum of all attentions $A_{sum} = \sum_{f=1}^{n-1}(A_y^f * \gamma_y^f) + A_y^b$ to equal to an all-ones matrix J_{256} with the same dimensions as the generated attention maps, 256×256 in our case. To facilitate this, we introduce a new loss term that aims to keep the attention from overlapping $L_{attention}$ defined as the Mean Absolute Error of A_{sum} and J_{256}:

$$L_{attention} = MAE(A_{sum}, J_{256})$$

The new term is added to the original AttentionGAN loss function:

$$L = L_{GAN} + \lambda_{cycle} * L_{cycle} + \lambda_{id} * L_{id} + \lambda_{attention} * L_{attention}$$

We picked $\lambda_{attention} = 10$ to put it in the correct range of the other selected hyperparameters.

For simplicity and future reference, we will name this model 'A-GAN 15 PAGE-Net'.

4.3.6 Impact of Domain Similarity over Convergence

When we compare the pizza and spaghetti bolognese classes, even though they look quite different, they have a lot of common features. First of all, they are quite often similar in shape, as most of the pizzas are round, although often there are triangular pieces or rectangular shaped pizzas in the images. Moreover, both food classes share a lot of colours, because of their common ingredients like tomato sauce and cheese. It was an interesting experiment to find out how much the similarity between two categories affect the convergence of the translation. We picked one of the most distant classes to the pizza class, which is cupcake. The majority of the images in the cupcake class are images of several cupcakes, not a single one (Figure: 4.5). This makes the shape really complex and a lot different from the pizza shape. Moreover, there are a lot of colours in the cupcakes, like colourful frostings, chocolate, fruits, jams and candies, all with different colours. This makes the translation a lot harder than the pizza to pasta translation.

We trained an AttentionGAN with fifteen content masks for 800 epochs. Some results are shown in Figure 4.11. We observe, that the algorithm has not been converged yet for sure. However in both directions of the translation a lot of features from the target class has been learned. In the pizza to cupcake direction, the algorithm has learned to generate quite well the frosting and the chocolate base of the cupcake, as well as some decorations on top. It is interesting to observe in the second example how for each mozzarella slice of the pizza, a separate cupcake has been generated.

In the other direction, from cupcake to pizza, the results are also interesting. The pizza texture has been generated well. However the shape needs to be improved. In the last example it has generated one big rectangular pizza on top of the whole box with cupcakes, where the chocolate frostings has become tomato sauce and the whipped cream has become mozzarella slices.

Clearly, the model needs more training. However it has a lot of potential. The large difference between the domain classes has affected a lot the convergence rate.

4.3.7 Convergence Metrics

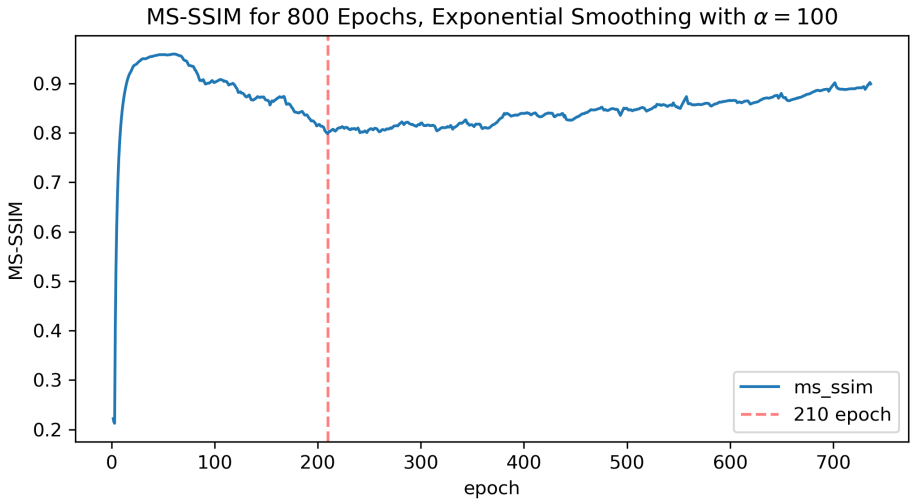
When working with GANs, it is useful to have a metric, which can evaluate when the convergence of the algorithm has been reached. Depending on the dataset and the domain of images, convergence could be reached on epoch 50, epoch 500, or even 1000. In our case the translation is really complex, as we are trying to change not only shape, texture or color, but all of them at once. Not only that, but dishes from the same type could be plated in completely different style, which complicates the problem even more. As shown above the number of epochs vary a lot when translating between different food categories. For this reason a convergence metric of the algorithm would be very valuable. We want to track the changes of the generated images during training and search for patterns that indicate convergence. The

FIGURE 4.11: Examples of pizza to cupcake translation on the left and cupcake to pizza on the right. The translations are from generated from an AttentionGAN with fifteen content masks, trained for 800 epochs.

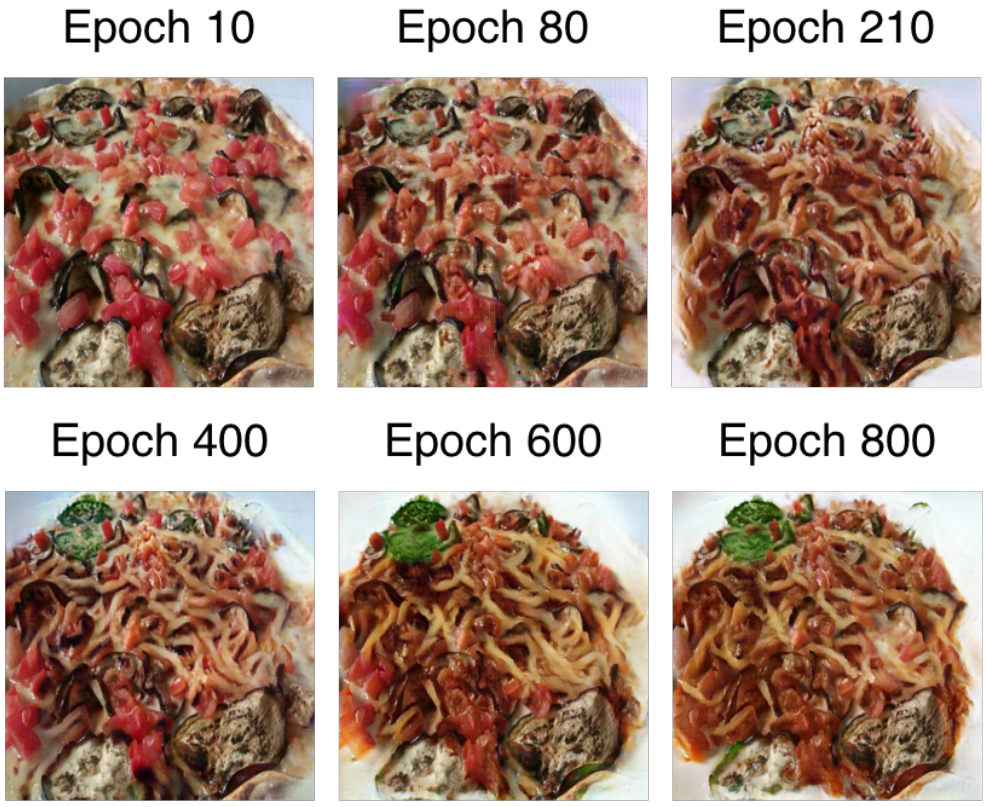
Multi-Scale Structural Similarity (MS-SSIM) index is a measure designed to assess the quality of an image [Wang, Simoncelli, and Bovik, 2003] by extracting structural information from it. In their work [Snell et al., 2017] and [Kancharla and Channappayya, 2018] the authors have shown that MS-SSIM is superior when used as a similarity metric between images than the usual pixel-wise distances like L_1 and L_2. In our work we employ MS-SSIM as a similarity metric for the change of texture across consecutive epochs. Figure 4.12 shows the MS-SSIM metric calculated on a small subset of the dataset while training A-GAN 15. In the initial epochs the gradients are high and there are a lot of changes. The first thing the algorithm learns is to consider everything as a background and outputs the input image which causes the huge spike of similarity going up to almost 1. Then the attention masks starts learning and around the 80 epoch it starts translating some parts of the image [4.12b]. This downward movement stops at the 210 epoch which coincides with a realistic image-to-image translation. From the 210 until the end the curve makes slow upward progress. However, we think that it has the potential of marking the point at which the images start translating to the other domain. After doubling the epochs after this turning point, the algorithm translates the image very well. From then on, one should run the algorithm until out of budget. Due to limited computational resources we were not able to run the algorithm for more than 800 epochs in order to observe whether it would go in some stationary state. For this reason we have not repeated the experiment with a different subset of the data data or different dataset to confirm its robustness. However, it deserves attention for future research.

4.3.8 Filtering

Our main goal is to generate good quality images, which ideally cannot be distinguished from real images. However, sometimes there are a lot of noisy images in the datasets which we would like to avoid when generating. When the algorithm picks a noisy image, which is completely out of the training domain, the attention generator of the AttentionGAN classifies the whole image as background [4.13]. Therefore, we use the attention mechanism for anomaly detection tool by measuring the percentage of the background pixels and filter those which are above certain threshold. In our experiments we found that on the problem of translating between pizza and pasta this threshold is 90%. Discarding these anomalies makes the algorithm more robust to anomalies or failed translations due to bad attention. However, since PAGE-Net searches for salient objects in the images it would fail to recognize it as an anomaly using the method we specified above. This feature has improved the evaluation metrics we have used to validate our results.



(A) The generated images at a key points in the training



(B) Plot of the MS-SSIM score training A-GAN 15 for 800 epochs.

FIGURE 4.12: On Figure 4.12a is displayed the smoothed MS-SSIM similarity metric tracking the change of texture across consecutive epochs. The lowest point marks the position where the generated images are start translating to the target domain. Not much progress is made in the generated images after doubling the epochs since entering the slow upward movement. Figure 4.12b depicts generated images on key points of the plot.

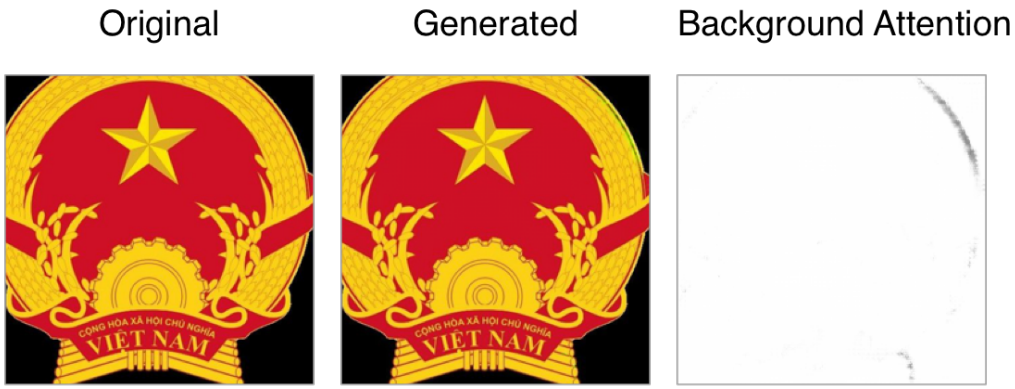


FIGURE 4.13: An example of anomaly detection when translating between pizza and pasta using the background attention.

Chapter 5

Results and Evaluation

In this section we show the results of the to promising approaches we experimented with: A-GAN 15 and A-GAN 15 PAGE-Net, along with CycleGAN as it is the most common approach in the literature for image-to-image food translation. All three models are trained on the pizza-to-pasta translation for 800 epochs. We compare them by evaluating their performance using visual inspection, quality of image metrics like Inception Score (IS) and Frechet Inception Distance (FID), whether they are correctly classified by a state of the art food classifier and whether they improve the classification accuracy when used as a data augmentation technique.

5.1 Visual Evaluation

First we start with the visual inspection of the generated images. Figure 5.1 illustrates the comparison of the A-GAN 15, A-GAN 15 PAGE-Net and CycleGAN over the pizza to pasta translation. Is can be observed that overall the A-GAN 15 has performed better than the others, generating the most realistic pasta images. However, CycleGAN produces good results as well, but the quality of the generations is lower. The lack of details in the shadows impedes the depth perception of the image. With regards to A-GAN 15 PAGE-Net, we observe that most of the time the results are noisier than in the other algorithms, although, in some cases it produces more realistic plates, as in the second row.

In the other direction, the pasta to pizza translation, appears to be more complicated problem (Figure: 5.2). Again A-GAN 15 performed the best, as the other two algorithms did not managed to generate the distinctive characteristics of pizza like the crust, mozzarella slices and cheese that well. At the same time, a lot of spaghetti texture has been left in the translated image.

On one hand, a big drawback of the A-GAN 15 PAGE-Net is that it fails to translate from smaller to bigger shapes. This is because the attention mechanism of PAGE-Net finds salient objects on the image, rather than concentrating on the parts of the image, that should be translated to the target domain. For example, it does not translate the dish of pasta meals into pizza crust, as the dish is considered as background.

On the other hand, the PAGE-Net has a more solid attention mask, which results in a better shaped generations, when we are converting a bigger meal to a smaller one (Figure 5.3). Notice that the pasta generations of A-GAN 15 PAGE-Net are more realistic in terms of a pasta presentation. Quite often they are served in the middle of a big bowl with a wide rim.

FIGURE 5.1: Comparison of translated pasta images. Overall A-GAN 15 generates more photo-realistic images than the others.

FIGURE 5.2

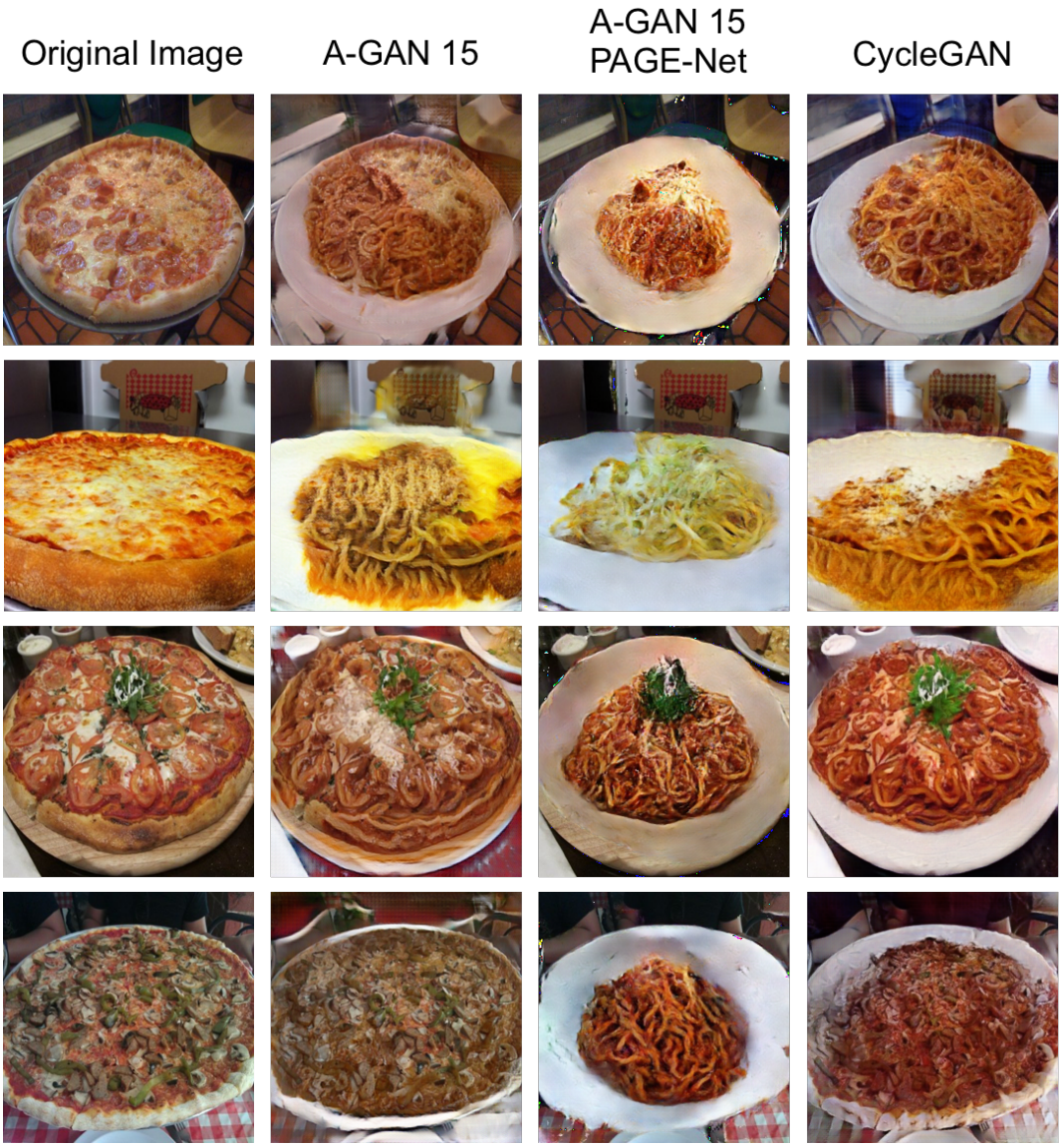


FIGURE 5.3: Examples of better translation from bigger to smaller meals (A-GAN 15 PAGE-Net)

5.2 Inception Score

The Inception score [Salimans et al., 2016] is a metric for evaluation of the GANs generated images. It is a commonly used metric designed to measure how realistic are the images according to an Inception classifier, trained on ImageNet. The classifier's function is to detect whether the images have a distinct object on them or not. This can be seen from the probability distribution of the outputs from the classifier. If a distribution is uniform, the images does not have any distinct object according to this classifier. Another thing which is useful is to measure the variety of the images we have generated, by summing a large number of label probability distributions. If there is a large variety of images in the generations, the ideal distribution this time is an uniform distribution. These two metrics are used to define the Inception score. The idea is that if we compare them, they should be as different as possible, therefore the score is defined by the difference between the two final distributions. Thus, the higher the score the better the images.

The inception score measured on our best performing models A-GAN 15 and A-GAN 15 PAGE-Net along with CycleGAN are shown in Table 5.1. Due to the lack of more state of the art results on this translation task we include the inception score of both the real images and the hold-out test data as a reference metric of what score is considered a good one. The highest inception score is that on the images from CycleGAN. We observe that despite the fact that visually A-GAN 15 has higher success rate in the translation, it has lower inception score than the A-GAN 15 PAGE-Net. Contrary to the expectation the real images has worse results than the generated ones. This might be due to the fact that this score varies a lot with the size of the data. For good evaluation the size of generation set should be large - 50000 images according to the authors, but in our case we have only 1500 generated images. Secondly, it is evaluating based on ImageNet's 1000 classes, which do not have a lot of food variety. Therefore, this score is not the most qualifying for our particular case.

5.3 Frechet Inception Distance

The Frechet Inception Distance (FID) score, on the other hand, is measuring how well the generated images are fitting in the real images distribution. The method uses the Inception classifier, like the Inception score. However, here the last layers of the NN are producing a multivariate Gaussian by calculating the mean and the covariance of the images. Then, the difference between the real data Gaussian and the fake data Gaussian are measured by using the Frechet distance, also called Wasserstein-2 distance. The output gives us the Frechet Inception Distance.

$$d^2((m_r, C_r), (m_f, C_f)) = \|m_r - m_f\|_2^2 + Tr(C_r + C_f - 2(C_r C_f)^{1/2}) \tag{5.1}$$

Here Tr sums all the elements of the diagonal, where m_r and C_r are the mean and the covariance of the real data, and respectively m_f and C_f - of the fake data.

As this score is not biased by the classifier's knowledge about food images, and it only measures the difference between Gaussians of the real and the fake data, the FID is way more appropriate and meaningful evaluation in the field of food image translations.

The FID scores are shown in [Table 5.2], alongside with the FID of the hold-out testing set of images for reference. The results confirm our visual observations that

Inception Score (IS)	
Method	Pizza and Pasta
A-GAN 15	2.88
A-GAN 15 PAGE-Net	3.06
CycleGAN	**3.10**
Real Training Set	2.20
Hold-out Set	1.82

TABLE 5.1: The inception score of the generated pizza and pasta samples. For this metric higher is better.

Frechet Inception Distance (FID)		
Method	Pizza to Pasta	Pasta to Pizza
A-GAN 15	**66.83**	**62.23**
A-GAN 15 PAGE-Net	112.43	74.558
CycleGAN	72.89	74.49
Hold-out Set	41.06	33.31

TABLE 5.2: The FID score of the pizza and pasta translations. For this metric lower is better.

A-GAN 15 has the highest success rate in generating photo-realistic images. Moreover, we consider its score as a very good one as it is not too far away from the reference hold-out set which contains real photos. When converting to pizza both CycleGAN and A-GAN 15 PAGE-Net have very similar scores. However, the big difference in the score between pizza-to-pasta and pasta-to-pizza of the A-GAN 15 PAGE-Net show that the algorithm performs much better at generating the dish around the pasta than the other way around. This is mainly due to its fixed background attention that focuses on the salient object in the source domain rather than the region that have to be changed to obtain the target domain. Depending on the use-case this might be a big limitation of the method.

5.4 Pre-trained Classifier Accuracy

Translating images from one category to another includes some bias in the shape of the generated images. For example, a triangular slice of pizza sometimes is translated into spaghetti with triangular shape. Therefore, it is valuable to investigate whether the fake images are realistic enough, despite the shape bias, to confuse a state of the art food classifier. The most accurate classifier trained on Food101, that is publicly available, is based on DenseNet-161 [Arka, 2019] and it has 93.26% top 1 accuracy and 99.01% top 5 accuracy on all 101 classes. We use it to classify the output of A-GAN 15 and applied filtering for anomalies and failed attentions, A-GAN 15 PAGE-Net and CycleGAN.

The results in tables 5.3 and 5.4 show that the generated images have high success rate in the translation to the target domain. The top 1 accuracy gives us the percentage of all images that have been classified as the target class with the highest confidence by the classifier and similarly for 'Top 5'. The last column shows the percentages of images which has been classified as the target class with a higher confidence, than the source class. This column is the most important one, as it shows the success rate of the translation to the target domain. In both directions A-GAN 15

is significantly superior than the other two methods with 79.9% and 68% successful translations. In the pizza-to-pasta translation the CycleGAN has slightly better results than A-GAN 15 PAGE-Net, however, in the other direction it performed much worse than the other two. A possible explanation of the lower success rate of the CycleGAN in the pasta-to-pizza translation is that a lot of the images have non-translated pasta texture. However, the top 5 accuracy is not that low meaning that they also contain a pizza-like texture. Overall, the results are quite satisfying, never the less, pasta-to-pizza translations are more successful.

An interesting observation is that A-GAN 15 generated pasta images have very high top 5 accuracy and success rate but a lot much lower top 1 accuracy. This means that the pizza to pasta translation was very successful but the classifier confuses it with some other food category. A possible future improvement is to train a multi-domain A-GAN 15 in which the discriminator would have to distinguish not only between pizza and pasta but with other categories as well. This would potentially improve the translation of the unique characteristics of each food category.

Accuracy on Generated Pasta Images			
Method	**Top 1**	**Top 5**	**Pasta > Pizza**
A-GAN 15	**62.7%**	**84.9%**	**79.9%**
A-GAN 15 PAGE-Net	42.9%	68.7%	64.0%
CycleGAN	47.1%	74.3%	65.9%

TABLE 5.3: Pre-trained classifier accuracy results on generated pasta images.

5.5 Extending Food Dataset with Synthetic Images

Neural networks require a lot of labelled training data, which is expensive to get most of the times. This is exactly the case with food recognition. The Food101 dataset consists of 1000 images per class, which is not so satisfactory, if we are targeting high accuracy, and it is almost impossible to train well-performing classifier without augmenting the data in some way.

An interesting idea for unusual data augmentation is to use the training set to train a GAN to generate more images using image-to-image translation. In this way, we are not only augmenting the data, by rotating or flipping the images, but we generate completely new samples, with background from other category images, which enlarge the diversity of the dataset.

In addition, this approach could be used as another evaluation of the generations, because if the accuracy of the test set does not drop significantly when the generated images are added to the original training dataset, this means that they are enough good not to confuse the training classifier.

Accuracy on Generated Pizza Images			
Method	**Top 1**	**Top 5**	**Pizza > Pasta**
A-GAN 15	**47.3%**	**75.9%**	**68.0%**
A-GAN 15 PAGE-Net	26.2%	60.0%	54.7%
CycleGAN	21.7%	59.2%	36.4%

TABLE 5.4: Pre-trained classifier accuracy results on generated pizza images.

Dataset	All classes	Pizza and Pasta	Pizza	Pasta
No generations	83.9%	**95.6%**	**97.6%**	93.6%
A-GAN 15	**85.3%**	93.0%	92.0%	**94.0%**
A-GAN 15 PAGE-Net	85.1%	93.1%	92.4%	**94.0%**
CycleGAN	84.2%	93.6%	93.6%	93.6%

TABLE 5.5: Test accuracy of a classifier trained after adding generations to the training set: The first results show the test accuracy of all 19 classes and the second is evaluation only on the pizza and spaghetti bolognese test classes, as these are the classes we added additional images to. The last two are evaluation only on the pizza class and only on the spaghetti class.

Just for the sake of the experiment we took a simple ResNet 50 classifier [He et al., 2016] pre-trained on ImageNet and trained it with 19 classes, randomly selected from the Food101 dataset - *apple pie, baby black ribs, baklava, connoli, cheesecake, chicken curry, churros, cup cake, filet mignon, french toast, paella, pizza, red velvet cake, seaweed salad, shrimp and grits, spaghetti bolognese, steak and waffles*. Firstly, we trained a classifier using only the original training set, without generations, to have a baseline. After that, we trained several classifiers, by including the generated images of our generative models into the original pizza and spaghetti-bolognese training set. The results are shown in Table 5.5. We can observe that the evaluation over all 19 classes has been improved by the images of each generative model. Therefore, it is this data augmentation technique is working. However, as we saw in the visual results, our pasta generations are more successful than the pizza ones. This is the reason that the pizza accuracy has dropped with when we included the generated images, but it appears that the pasta accuracy has been improved.

To sum up, the synthetic images appear to be good enough to help a classifier identify distinctive features, even though not all of them can deceive a person.

Chapter 6

Conclusions and future work

In our Master Thesis project we proposed applying attention in food image-to-image translation, using AttentionGAN and PAGE-Net. We propose two different models and compare them along with the current state of the art method - CycleGAN.

In the beginning, we are looking at the previous work that have been done in this area, discuss what are the drawbacks of the methods proposed and how can we overcome them. We follow up with the detailed explanation of the image-to-image translation problem, AttentionGAN architecture, differences between its versions and the concept of salient object recognition using PAGE-Net.

The next part is the experimental, where we define the models that we have experimented with and how they performed. During the experiments we selected the two best performing models: A-GAN 15 and A-GAN 15 PAGE-Net. Both models are enhancements of the base AttentionGAN, one of with is using only the attention mechanism of the base AttentionGAN, however we concluded that adding more content masks to it improved the convergence rate and the overall generations. That is the reason we decided to add 5 more content masks to the base AttentionGAN and create the A-GAN 15. Moreover, the A-GAN 15 architecture has an option to apply filtering when generating synthetic images that removes failed translations due to anomalies in the data. The other model, A-GAN 15 PAGE-Net, is based on A-GAN 15 but using the PAGE-Net algorithm to produce the background mask, while the foreground masks are scaled by training parameters. We added an additional loss function to the model as well, which helped with the adjustments of the foreground masks. Moreover, we propose using MS-SSIM as a convergence metric for attention based generators by measuring the texture similarity of a subset of the dataset between each epoch in order to find out when the Attention GAN starts producing successful translations.

In the end, we produce very good results of photo-realistic food images. We used several metrics to evaluate the generations. The IS and FID scores of the two methods are compared along with the CycleGAN. Two interesting evaluations were made after that. The first one answers the question if the generated images are good enough to fool the state of the art Food101 classifier. The second one is using the generations as a data augmentation technique and measures if the test accuracy of a classifier will be higher if it is trained using these images as augmentation. All validations were showing good results, which concludes that indeed a lot of the generations were successful. In both visual and quantitative evaluation A-GAN 15 excels over the other methods beating the current state of the art architecture for food-to-food translation.

Some limitations we have faced:

- Despite the loss function that we added in the A-GAN 15 PAGE-Net model, sometimes a little amount of noise is appearing. More detailed research needs to be done in order to fix this problem completely.
- A big drawback of PAGE-Net is that it does not generate good images if the source meal is small and the target meal is large, as it considers the area of the small image as a foreground and it cannot be extended to occupy a larger area.
- AttentionGAN is creating really realistic images, however the ones that are failing are bad because of the attention, which needs to be improved.
- If the generation is used as a data augmentation, a drawback is that the generator does not add noise and it cannot generate images forever. For one pizza image it can generate only one pasta image and the other way around.

Future work:

- We can improve the attention of the AttentionGAN by adding a loss function, which guides the different attention to focus on different features of the image. For example, one attention which can detect cheese, another one which can detect tomatoes and so on.
- Another improvement is to extend the GAN to generate multiple domains images. As already discussed this has a potential of improving the texture of the generated images.
- The convergence metrics that we have introduced, can be researched and improved, possibly by using a simpler dataset on which it's easier to validate that the algorithm has converged.

Appendix A

Distribution of work

We are two team members Mariya Mladenova and Petar Tonchev. We have done together almost the whole project following pair programming techniques, however, each of us had a focus on some specific aspects of the project.

Petar Tonchev:

- AttentionGAN experimentation
- Additional content masks integration
- Integrate the γ scaling factor of the A-GAN 15 PAGE-Net
- Convergence metrics
- Inception score and FID metrics
- CycleGAN experimentation
- Lead speaker in the weekly discussions with our supervisors
- Research and development of various experiments not included in the report

Mariya Mladenova:

- PAGE-Net integration
- Experimentation with PAGE-Net integration alternatives
- Integrate the γ scaling factor of the A-GAN 15 PAGE-Net
- additional loss function of A-GAN 15 PAGE-Net
- Evaluation with the pre-trained Food101 classifier
- Evaluation with the generations used as data augmentation
- Pizza to cupcake experiment
- Research and development of various experiments not included in the report

Bibliography

Aguilar, Eduardo et al. (2018). "Grab, pay, and eat: Semantic food detection for smart restaurants". In: *IEEE Transactions on Multimedia* 20.12, pp. 3266–3275.

Arka George Christopoulos, Manisha Oladimeji Mudele Prakhar Tripathi Stark (2019). *Food Classification with DenseNet-161*. https://github.com/Escanor1996/food_classifier_deployment.

Bolaños, Marc, Aina Ferrà, and Petia Radeva (2017). "Food ingredients recognition through multi-label learning". In: *International Conference on Image Analysis and Processing*. Springer, pp. 394–402.

Bossard, Lukas, Matthieu Guillaumin, and Luc Van Gool (2014). "Food-101–mining discriminative components with random forests". In: *European conference on computer vision*. Springer, pp. 446–461.

Brock, Andrew, Jeff Donahue, and Karen Simonyan (2018). "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096*.

Choi, Yunjey et al. (2018). "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797.

Gokaslan, Aaron et al. (2018). "Improving shape deformation in unsupervised image-to-image translation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 649–665.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Han, Fangda, Ricardo Guerrero, and Vladimir Pavlovic (2020). "CookGAN: Meal Image Synthesis from Ingredients". In: *The IEEE Winter Conference on Applications of Computer Vision*, pp. 1450–1458.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Kancharla, Parimala and Sumohana S Channappayya (2018). "Improving the visual quality of generative adversarial network (GAN)-generated images using the multi-scale structural similarity index". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3908–3912.

Li, Yijun et al. (2017). "Universal style transfer via feature transforms". In: *Advances in neural information processing systems*, pp. 386–396.

Lu, Yuzhen (2016). "Food image recognition by using convolutional neural networks (cnns)". In: *arXiv preprint arXiv:1612.00983*.

Marin, Javier et al. (2019). "Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Mirza, Mehdi and Simon Osindero (2014). "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784*.

Nakano, Kizashi et al. (2019a). "DeepTaste: Augmented reality gustatory manipulation with GAN-based real-time food-to-food translation". In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, pp. 212–223.

Nakano, Kizashi et al. (2019b). "Enchanting Your Noodles: GAN-based Real-time Food-to-Food Translation and Its Impact on Vision-induced Gustatory Manipulation". In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, pp. 1096–1097.

Papadopoulos, Dim P et al. (2019). "How to make a pizza: Learning a compositional layer-based GAN model". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8002–8011.

Perarnau, Guim et al. (2016). "Invertible conditional gans for image editing". In: *arXiv preprint arXiv:1611.06355*.

Sabini, Mark, Zahra Abdullah, and Darrith Phan (2018). "GAN-stronomy: Generative Cooking with Conditional DCGANs". In:

Salimans, Tim et al. (2016). "Improved techniques for training gans". In: *Advances in neural information processing systems*, pp. 2234–2242.

Salvador, Amaia et al. (2017). "Learning cross-modal embeddings for cooking recipes and food images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3020–3028.

Snell, Jake et al. (2017). "Learning to generate images with perceptual similarity metrics". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 4277–4281.

Tang, Hao et al. (2019). "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks". In: *arXiv preprint arXiv:1911.11897*.

Tanno, Ryosuke et al. (2018). "Magical Rice Bowl: A Real-time Food Category Changer". In: *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1244–1246.

Wang, Hao et al. (2019a). "Learning cross-modal embeddings with adversarial networks for cooking recipes and food images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11572–11581.

Wang, Wenguan et al. (2019b). "Salient object detection with pyramid attention and salient edges". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1448–1457.

Wang, Zhou, Eero P Simoncelli, and Alan C Bovik (2003). "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. Ieee, pp. 1398–1402.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer". In: *arXiv preprint arXiv:1612.03928*.

Zhang, Han et al. (2018). "Stackgan++: Realistic image synthesis with stacked generative adversarial networks". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8, pp. 1947–1962.

Zhu, Jun-Yan et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.